METODI ITERATIVI STAZIONARI

$Ax = b$ $\quad A \in \mathbb{C}^{n\times n}$, $x, b \in \mathbb{R}^n$ $\quad x^* = A^{-1}b$

Si scrive $A = M - N$ con M non singolare (nella pratica "facile da invertire")

$Mx = Nx - b \rightsquigarrow x = M^{-1}Nx + M^{-1}b = Bx + c$

Si definisce $x^{k+1} = Bx^k + c$ $\quad k \geq 0$ (partendo da un certo x_0 tipicamente 0)

Si dicono metodi STAZIONARI perché la matrice di iterazione B è sempre la stessa.

Osserviamo che chiaramente vale $x^* = Bx^* + c$, inoltre

$$e^k = x^k - x^* \rightsquigarrow e^{k+1} = Be^k = B^2e^{k-1} = B^ke^0$$

e dunque il metodo converge (indipendentemente da e_0) sse

$$\lim_{k\to\infty} B^k = 0 \quad \text{sse} \quad \rho(B) < 1$$

OSS: Se posso scomporre $B = S\Lambda S^{-1}$ (autoval.) la condizione è evidente dato che

$$B^k = S\Lambda^k S^{-1}$$

con $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ e quindi $B^k \to 0$ sse $(\max_i \lambda_i)^k \to 0$ sse $\lambda_{max} < 1$.
(Nel caso generale non lo dimostro ma vale lo stesso)

ESEMPI: (di metodi) In generale abbiamo:

$A = D - L - U$ con $D = \text{diag}(a_{11}, \dots, a_{nn})$

$$L = -\begin{bmatrix} 0 & & 0 \\ a_{ij} & \ddots & \\ & & 0 \end{bmatrix}$$

$$U = -\begin{bmatrix} 0 & & a_{ij} \\ & \ddots & \\ 0 & & 0 \end{bmatrix}$$

Jac • Jacobi: $x_i^{(k+1)} = \dfrac{\left(b_i - \sum_{j\neq i} a_{ij}x_j^{(k)}\right)}{a_{ii}}$

si usa $M = D$ e $N = L + U$ $\rightsquigarrow B = M^{-1}N = D^{-1}(L+U)$.

Converge se DDS o DDD + irriducibile.
(Quindi nei nostri casi converge, peccato che sia lentissimo, quindi è meglio CG (Pb ellittici))

GS • Gauss-Seidel: $x_i^{(k+1)} = \dfrac{\left(b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right)}{a_{ii}}$

si usa $M = D - L$ e $N = U$ $\rightsquigarrow B = (D-L)^{-1}U$

Converge per A sdp ma è cmq lenta!

SOR • Si introduce un parametro ω:

$M_\omega = \omega^{-1}D - L$

$N_\omega = (1-\omega)\omega^{-1}D + U$

dal punto di vista dei conti non cambia, ma così abbiamo da giocarci un parametro per aumentare la velocità di convergenza.

Cerchiamo ω^* parametro ottimo t.c.

$$\rho(B_{\omega^*}) = \min_{\omega>0} \rho(B_\omega)$$

Abbiamo che se A sdp tridiagonale o tridiagonale a blocchi allora Jacobi e Gauss-Seidel convergono, GS più velocemente di Jacobi (ma sono entrambi lenti). Anche SOR converge e si ha

$$\omega^* = \frac{2}{1+\sqrt{1-\rho(B_J)}} \in (1,2)$$

B_J matrice di Jacobi

(che però non è ovvio da calcolare).

Il grafico di $\rho(B_\omega)$ al variare di ω è tipo:

quindi conviene approssimare ω^* per eccesso nel caso non riusciamo a calcolarlo (così cresce meno velocemente).

In ogni caso anche il SOR perde contro il CG perché la convergenza è simile ma in CG non devo calcolare nessun parametro!

Allora li buttiamo via? NO, faranno parte dei metodi MULTIGRID.

SIMPLE ITERATION

E' equivalente al precedente:

$$B = M^{-1}N = I - M^{-1}A$$

$N = M - A$

Quindi riparto da capo e vedo le cose in un'ottica di precondizionamento.

$Ax = b$ ma scelgo un precondizionatore M t.c. $M^{-1}A \approx I$ (*)

in qualche senso che dipende dal metodo

Ci si aspetta che

$M^{-1}(b - Ax^k)$ approssimi $e^k = x^* - x^k$.

$b - Ax^k = r^k$

Infatti:

$$M^{-1}(b - Ax^k) = M^{-1}(Ax^* - Ax^k) = M^{-1}Ae^k \overset{\text{se (*)}}{\approx} e^k. \quad (\star)$$

Viene naturale considerare:

$$x^{k+1} = x^k + M^{-1}(b - Ax^k) \quad (\bullet)$$

infatti se fosse $M^{-1}A = I$ non in senso approssimato cioè detto meglio se volesse (*) con l'uguale, avremmo

$$x^{k+1} = x^k + e^k = x^k + x^* - x^k = x^* \quad \text{☺}$$

Questo metodo dal punto di vista algebrico è equivalente a quello di prima infatti ho:

$(\bullet) \longrightarrow x^{k+1} = (I - M^{-1}A)x^k + M^{-1}b = M^{-1}Nx^k + M^{-1}b$ (splitting)

$I - M^{-1}A = B$

ALGORITMO (per un qualsiasi metodo di SPLITTING)

Dato x^0, calcolo $r^0 = b - Ax^0$ e risolvo $Mz^0 = r^0$ (invece non calcolo davvero M^{-1} ☺)

per $k \geq 1$ pongo $x^k = x^{k-1} + z^{k-1}$

$r^k = b - Ax^k$

risolvo $Mz^k = r^k$ ← risolvo esattamente

$k = k+1$

← da qualche parte c'è un test d'arresto, o subito o prima di risolvere $Mz^k = r^k$ (che è la cosa pesante)

MA, MI SERVE VERAMENTE? NO → posso usare un metodo iterativo anche qui

Mi rendo conto che posso non calcolare esattamente z^{k+1}, visto che se sto chiedendo x con una precisione, non ho senso trovare z^k con precisione maggiore!

In realtà mi interessa essere poco precisa nelle prime iterazioni, ovvero quando sono ancora lontano dalla soluzione "giusta", ma essere più precisa dopo.

Ho un OUTER ITERATION (quello per trovare x) con una certa condizione d'arresto (CA) e un INNER ITERATION (quello per trovare z^k) con una CA che dipende dalla CA dell'outer iteration.

Se non calcolo esattamente z^k il metodo si dice METODO ITERATIVO INESATTO.

Ora abbiamo il problema, dato $A=M-N$ splitting, qual'è M candidato a essere precondizionatore per PCG?

Una condizione che sappiamo deve essere soddisfatta è M sdp (quindi se in realtà nella pratica a volte si usa $M=M'+R$ con M' sdp e R "piccola" e se R non cambia di molto M' funziona lo stesso).

- Un possibile candidato è quello di "derivare" M dal metodo di Jacobi: $M=D=\text{diag}(A)$

  Pb ellittico $\rightsquigarrow A$ sdp $\Rightarrow a_{ii}>0$

- SOR simmetrico (SSOR) (GS = SOR per $\omega=1$)

  Partiamo da GS, abbiamo: $M=D-L$, $N=U$

  alternativa: $M=D-U$, $N=L$

  (tanto posso partire da $i=N$ e tornare indietro al posto che da $i=1$ e andare avanti)

  Quindi in generale per il SOR possiamo fare

  $$M_\omega=\omega^{-1}D-L \qquad N_\omega=(1-\omega)\omega^{-1}D+U \qquad \text{(SOR)}$$

  $$\tilde{M}_\omega=\omega^{-1}D-U \qquad \tilde{N}_\omega=(1-\omega)\omega^{-1}D+L \qquad \text{(BACKWARD SOR)}$$

  L'SSOR consiste nel fare un passo di SOR e uno di backward SOR:

  $$\hat{x}^{k+1}=M_\omega^{-1}(N_\omega x^k+b)$$

  $$x^{k+1}=\tilde{M}_\omega^{-1}(\tilde{N}_\omega \hat{x}^{k+1}+b)$$

  Nell'implementazione questi due passi vengono mantenuti separati, ma per vedere la matrice di iterazione del metodo possiamo scriverla in un solo passo (facendo dei brutti conti)

  $$x^{k+1}=\underbrace{M_\omega^{-T}N_\omega^T M_\omega^{-1}}_{\hat{M}^{-1}}N_\omega x^k+M_\omega^{-T}(N_\omega^T+M_\omega)M_\omega^{-1}b$$

  dove abbiamo considerato il caso $A=A^T$ e quindi $M_\omega^T=\tilde{M}_\omega$ e $N_\omega^T=\tilde{N}_\omega$.

  Osserviamo che $\hat{M}^{-1}=M_\omega^{-T}N_\omega^T M_\omega^{-1}$ è simmetrica (come richiesto).

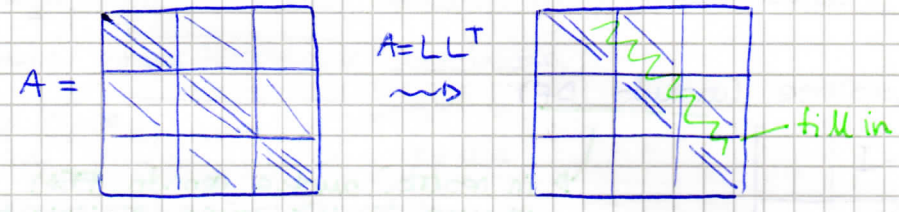
  Si può verificare che:

  $$A=\bar{M}_\omega-\bar{N}_\omega$$

  con $\bar{M}_\omega=\frac{\omega}{2-\omega}M_\omega D^{-1}M_\omega^T$. (Così vedo che $\bar{M}_\omega$ non solo è simmetrica ma è anche sdp, che era l'altra condizione richiesta)

- C'è un ultimo precondizionatore generico (ovvero che non richiede un'analisi degli autovalori del problema e che posso usare per tutti).
  PRECONDIZIONATORE nella FORMA INCOMPLETA di CHOLESKY.

Abbiamo il problema $Ax=b$, supponiamo per esempio di essere nel caso delle diff. finite e $-\Delta u = f$ abbiamo



Si prende $\tilde{L}$ che è quella "non riempita" avero con gli zeri dove A aveva gli zeri e $=L$ dove $A \neq 0$. Prendiamo

$$M = \tilde{L}\tilde{L}^T \neq A$$

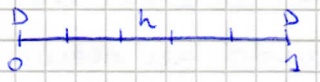
($\tilde{L}$ si calcola con ichol in matlab)

MATRICI ASSOCIATE AL PROBLEMA del LAPLACIANO

5/12/2012 (17)

Consideriamo il problema:

$$\begin{cases} -u'' = f & \text{su } \Omega = (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

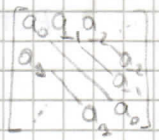


N = # sottointervalli
$N+1$ nodi
$h = 1/N$

utilizzando le differenze finite approssimiamo $-u''(x)$:

$$-u''(x) \cong \frac{1}{h^2}\left[-u(x-h) + 2u(x) - u(x+h)\right.$$

$$\rightsquigarrow A_n = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \qquad n = N-1$$



Abbiamo che A_n è tridiagonale ed è anche una matrice di Toeplitz, ovvero ha elementi costanti lungo le diagonali

Posso interpretare gli a_k nella matrice di Toeplitz come i coefficienti di fourier di una funzione f (FUNZIONE GENERATRICE di A_n)

$$a_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)\, e^{-itk}\, dt$$

e quindi ho una corrispondenza tra matrici Toeplitz e funzioni (serie di fourier) cioè se ho gli a_k posso trovare f

$$f(t) = \sum_{\mathbb{Z}} a_k e^{itk}$$

Quindi nel nostro caso, $A_n = \text{tridiag}(-1, 2, -1)$

$f(t) = 2 - e^{-it} - e^{it} = 2 - 2\cos t$ funzione generatrice di A_n

Abbiamo che:

$$\lambda_s(A_n) = f(x_s) \quad \text{con} \quad x_s = \frac{s\pi}{n+1} \qquad s = 1, \dots, n$$

$$= 2 - 2\cos\left(\frac{s\pi}{n+1}\right) = 4\,\text{sen}^2\left(\frac{s\pi}{2(n+1)}\right)$$

Quindi vediamo che nel nostro caso $\lambda_1 (= \lambda_{min})$ va a zero come h^2 (sostituiamo $n+1 = N = 1/h$), mentre $\lambda_n (= \lambda_{max})$ rimane limitato (≤ 4) e quindi:

$$K_2(A_n) \cong h^{-2}$$

A_n è anche una τ ovvero diagonalizzata da trasformazioni di → ALGEBRA $A = V\Lambda_n V^{-1}$ $A_n = S\Lambda_n S$ CIRCOLANTI $A_n = F_n \Lambda_n F_n^{-1}$ F_n trasf discreta fourier

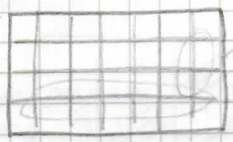
di tipo 1. Quindi

$$A_n = S_n \Lambda S_n$$

con S_n ortogonale, simmetrica $S_n = \sqrt{\frac{2}{n+1}}\left[\text{sen}\left(\frac{sr\pi}{n+1}\right)\right]_{s,r=1}^{n} = [S_1 | \dots | S_n]$

Possiamo ora al caso 2 dimensionale, quindi abbiamo il problema

$$\begin{cases} -\Delta u = f & \text{su } \Omega = (0,1)\times(0,1) \\ u|_{\partial\Omega} = 0 \end{cases}$$



n_2 = # righe della griglia

n_1 = # nodi per riga

ottenuto la matrice (usando DF)

$$A_n = \begin{bmatrix} 4 & -1 & & -I & & \\ -1 & \ddots & \ddots & & & \\ & \ddots & 4 & & & \\ -I & & & 4 & -1 & -I \\ & & & -1 & \ddots & \ddots \\ & & & -I & \ddots & 4 \end{bmatrix}$$

↳ in realtà anche usando FEM è una triangolazione strutturata uniforme

(ho solo 4 componenti per riga non nulle perché 2 vengono 0 per ortogonalità dei gradienti)

tridiagonale a blocchi con blocchi tridiagonali.
Vediamo qual'è la funzione generatrice, viene:

$$f(u,v) = 4 - 2\cos u - 2\cos v$$

infatti considerando gli elementi di A_n abbiamo

$$f(u,v) = 4 - e^{i(-1)u}e^{i0v} - e^{i1u}e^{i0v} - e^{i0u}e^{i(-1)v} - e^{i0u}e^{i1v}$$

(che è quella sopra).

Posso separare la u e la v e scrivere $f(u,v) = (2-2\cos u) + (2-2\cos v)$ quindi posso pensare di separare i campionamenti su u e v (per il calcolo degli autovalori).

Autovalori: $\lambda^{(A_n)}_{(s_1,s_2)} = f\left(\frac{\pi s_1}{n_1+1}, \frac{\pi s_2}{n_2+1}\right)$ $\quad s_1 = 1,\dots,n_1$ $\quad s_2 = 1,\dots,n_2$

(sto numerando la griglia con notazione a doppio indice)

$$\| \quad \lambda_k(A_n) \quad \text{con} \quad k = s_1 + (s_2-1)n_1$$

se penso all'indicizzazione "totale"

Abbiamo due in questo caso:

$\lambda_{max} \leq 8$

$\lambda_{min} \geq h^2 \longrightarrow K_2(A_n) \cong O(h^{-2})$.

A_n è ancora una $\tau \longrightarrow A_n = S_n \Lambda S_n$ con

$S_n = S_{n_1} \otimes S_{n_2}$ con S_{n_i} Trasformata discreta di seni di tipo I (1)

def: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{m\times m}$ $\quad A\otimes B = [a_{ij}B]_{mn\times mn} = \begin{bmatrix} a_1B & a_2B & a_3B \\ a_4B & a_5B & \end{bmatrix}$

e quindi:

$$S_{n_1 n_2} = \sqrt{\frac{2}{n_1+1}}\sqrt{\frac{2}{n_2+1}}\left[\left[\operatorname{sen}\left(\frac{s_1\pi t_1}{n_1+1}\right)\right]_{s_1=1,\dots,n_1} \otimes \left[\operatorname{sen}\left(\frac{s_2\pi t_2}{n_2+1}\right)\right]_{s_2=1,\dots,n_2}\right]$$

PROP: (PROPRIETÀ PRODOTTO TENSORIALE) $A,B,C \in \mathbb{C}^{2\times 2}$ = cambio nelle prop

- (i) $A\otimes(B+C) = A\otimes B + A\otimes C$
- (ii) $(A+B)\otimes C = A\otimes C + B\otimes C$
- (iii) $(\alpha A)\otimes B = A\otimes \alpha B = \alpha(A\otimes B)$ $\quad \alpha \in \mathbb{C}$
- (iv) $A\otimes(B\otimes C) = (A\otimes B)\otimes C$
- (v) $(A\otimes B)^H = A^H \otimes B^H$
- Se $A,C \in \mathbb{C}^{n\times n}$, $B,D \in \mathbb{C}^{m\times m}$ $\quad (A\otimes B)(C\otimes D) = (AC)\otimes(BD)$
- $A\otimes B$ non singolare se A, B non singolari e $(A\otimes B)^{-1} = A^{-1}\otimes B^{-1}$
- $\exists\, \Pi \in \mathbb{R}^{nm\times nm}$ di permutazione t.c. $\Pi(A\otimes B)\Pi^{-1} = B\otimes A$
- Se $A \in \mathbb{C}^{n_1\times n_1}$ e $B \in \mathbb{C}^{n_2\times n_2}$ allora: $\lambda_k(A\otimes B) = \lambda_{i_1}(A)\cdot\lambda_{i_2}(B)$ $\quad k = 1,\dots,n_1n_2$, $i_s = 1,\dots,n_s$

e gli autovettori sono il prodotto tensoriale di quelli di A e B.

- $\lambda_k(I_{n_2}\otimes A + B\otimes I_{n_1}) = \lambda_{i_1}(A)+\lambda_{i_2}(B)$ con A, B come prima.

ESEMPIO: APPLICAZIONE ~~TEOREMA~~ AUTOVALORI di MATRICI HERMITIANE

Vogliamo considerare il problema:

$$\begin{cases} u^{IV} = f & \text{in } (0,1) \\ u^{(j)}(0) = u^{(j)}(1) = 0 & j = 0,1 \end{cases}$$

con le differenze finite ottengo

$$\tilde{A}_n = \begin{bmatrix} 6 & -4 & 1 & 0 \\ -4 & \ddots & \ddots & 1 \\ 1 & \ddots & \ddots & -4 \\ 0 & 1 & -4 & 6 \end{bmatrix}$$

che deriva dal fatto che ho:

$$u^{IV}(x) \cong \left[u(x-2h) - 4u(x-h) + 6u(x) - 4u(x+h) + u(x+2h)\right] \cdot \frac{1}{h^4}$$

questa formula posso ricavarla in 2 modi, o facendo la serie di Taylor e tutti i conti o considerando:

$$u^{IV} = -(-u'')''$$

sappiamo che $u''(x) \approx \frac{1}{h^2}\left[u(x-h) - 2u(x) + u(x+h)\right]$, poniamo $v(x) = -u''(x)$

$$u^{IV} = -v''(x) \approx -\frac{1}{h^2}\left[v(x-h) - 2v(x) + v(x+h)\right] =$$

$$= +\frac{1}{h^2}\left[+u''(x-h) \mp 2u''(x) + u''(x+h)\right]$$

$$= \frac{1}{h^4}\left[u(x-2h) - 2u(x-h) + u(x) - 2\left(u(x-h) - 2u(x) + u(x+h)\right) + \cdots \right.$$
$$\left. + u(x) - 2u(x+h) + u(x+2h)\right] =$$

$$= \frac{1}{h^4}\left[u(x-2h) - 4u(x-h) + 6u(x) - 4u(x+h) + u(x+2h)\right]$$

Mi chiedo, questo trucco funzionerà anche per il calcolo delle funzioni generatrici?

$$-u'' \rightsquigarrow f(t) = 2 - 2\cos t$$

$$u^{IV} \rightsquigarrow \tilde{f}(t) = 6 - 4e^{it} - 4e^{-it} + e^{2it} + e^{-2it} =$$
$$= 6 - 8\cos t - 2\cos(2t)$$

Si vede che $\tilde{f}(t) = f^2(t)$. Quindi riesco a trovare una relazione tra gli autovalori di $\tilde{A}_n$ (u^{IV}) e di A_n ($-u''\ldots$)? (visto che li calcolo come campionamento di f e $\tilde{f}$)

Vediamo la matrice $\tilde{A}_n$ in questa forma

$$\tilde{A}_n = \begin{bmatrix} 5 & -4 & 1 & & \\ -4 & 6 & \ddots & \ddots & \\ 1 & \ddots & \ddots & 1 & \\ & \ddots & 6 & -4 & \\ & & 1 & -4 & 5 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & & 1 \end{bmatrix} \leftarrow \text{matrice di correzione, ha rango 2, è sdp}$$

$$= A_n^2 + e_1 e_1^T + e_n e_n^T$$

A_n^2 ← si vede facendo il quadrato di A_n

Vogliamo ricondurci al teorema CF, consideriamo il quoziente di Rayleigh

$$r_{\tilde{A}_n}(x) = \frac{x^T \tilde{A}_n x}{x^T x} = \underbrace{x^T A_n^2 x}_{\geq 0} + \underbrace{x^T e_1 e_1^T x + x^T e_n e_n^T x}_{\geq 0} \geq r_{A_n^2}(x)$$

e quindi per Courant-Fischer $\lambda_j(\tilde{A}_n) \geq \lambda_j(A_n^2) = \left(2 - 2\cos\left(\frac{j\pi}{n+1}\right)\right)^2$

e dunque l'autovalore minimo, che si ottiene per $j=1$, è

$$\geq \left(4\sin^2\left(\frac{\pi}{2(n+1)}\right)\right)^2 \simeq \frac{\pi^4}{n^4}$$

e dato che il λ_{max} è limitato si ottiene che

$$K_2(\tilde{A}_n) \text{ è al più } O(n^4).$$

Ora ci troviamo nel caso di correzione di rk 1 (Interallacciamento):

A $\quad B = A + \sigma_1 u_1 v_1^H \quad C = B + \sigma_2 u_2 v_2^H$ ($C = \tilde{A}_n$)

A_n^2

i teoremi visti ci dicevano che

N.B. in questi teoremi gli autovalori sono indicizzati al contrario

$$\lambda_{j+1}(A) \geq \lambda_j(B) \geq \lambda_j(A)$$
$$\lambda_{j-1}(B) \geq \lambda_j(C) \geq \lambda_j(B) \qquad j = 2, \dots, n$$

e mettendo insieme le due relazioni ho:

$$\lambda_{j+2}(A) \geq \lambda_{j+1}(B) \geq \lambda_j(C) \geq \lambda_j(B) \geq \lambda_j(A) \qquad j = 3, \dots, n$$

ovvero ho che controllo tutti gli autovalori di C ($\tilde{A}_n$) con gli autovalori di A (A_n^2) tranne λ_1 e λ_2, (che però in questa notazione sono i più grandi).

In realtà ora mostriamo con un argomento di immersione che possiamo controllare anche λ_1 e λ_2.

Considero A_{n+2}^2

$$A_{n+2}^2 = \begin{bmatrix} 5 & -4 & 1 & & \\ -4 & & & & \\ 1 & & \tilde{A}_n & & \\ & & & -4 & 1 \\ & & & 5 & -4 \\ & & & -4 & 5 \end{bmatrix}$$

avvero immergo $\tilde{A}_n$ in A_{n+2}^2.

sono le dimensioni delle matrici

A_{n+2} $\quad B_{n+1}$ $\quad C_n$

$\uparrow_2$ $\quad \updownarrow$

A_{n+2}^2 $\quad A_n$

Teo di proiezione $\longrightarrow$ $\lambda_j(A) \geq \lambda_j(B) \geq \lambda_{j+1}(A)$ $\quad j = 1, \dots, m-1$ $\quad m = n+2$

$$\lambda_j(B) \geq \lambda_j(C) \geq \lambda_{j+1}(B) \qquad j = 1, \dots, m-2$$

Componendo come prima:

$$\lambda_j(A) \geq \lambda_j(C) \geq \lambda_{j+1}(B) \geq \lambda_{j+2}(A) \qquad j = 1, \dots, \underbrace{m-2}_{=n}$$

e quindi controllo tutti gli autovalori di C ($= \tilde{A}_n$).

11/12/2012

METODI MULTIGRID

Partiamo dal problema unidimensionale

$$\begin{cases} -u'' = f & \text{su } (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

Tramite DF o FE otteniamo la matrice $A_n = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$

Vediamo cosa succede se applichiamo Jacobi alla risoluzione del nostro sistema lineare.

Abbiamo $\quad A = D - L - U \quad M = D \quad N = L + U$

Matrice di iterazione: $B_J = D^{-1}(U+L)$

Introduciamo un metodo di Jacobi pesato:

$$x^{k+1} = B_{J,\omega} x^k + \omega \underbrace{D^{-1} b}_{=c_J}$$

con $B_{J,\omega} = (1-\omega) I + \omega B_J$.

Abbiamo

$$\begin{aligned} B_{J,\omega} &= (1-\omega) I + \omega D^{-1}(L+U) = \\ &= (1-\omega) I + \omega D^{-1}(D-A) = \\ &= (1-\omega) I + \omega I - \omega D^{-1} A = \\ &= I - \omega D^{-1} A \end{aligned}$$

Ricordiamo che stiamo usando $N+1$ nodi equispaziati, $h = 1/N$, $n = N-1$

Abbiamo che nel nostro caso:

$$B_{J,\omega} = I - \frac{\omega}{2} \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$$ simmetrica.

$$\lambda_i(B_{J,\omega}) = 1 - \frac{\omega}{2} \lambda_i(A)$$ e sappiamo che $\lambda_i(A) = 2 - 2\cos\left(\frac{i\pi}{N}\right) \overset{!}{=} 4 \operatorname{sen}^2\left(\frac{i\pi}{2N}\right)$ $i = 1, \dots, n = N-1$

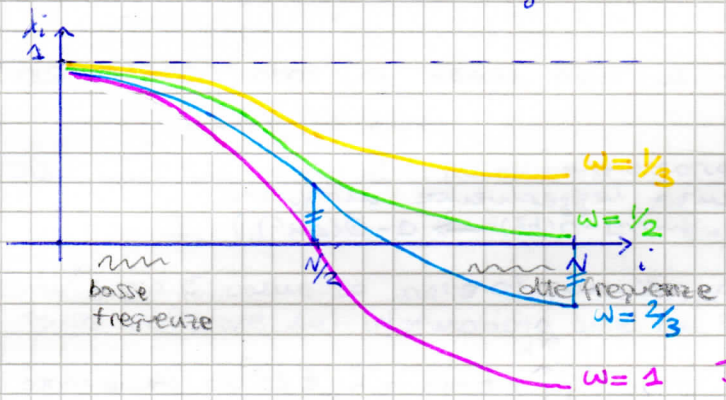
$$w^{(i)} = \left[\operatorname{sen}\left(\frac{ij\pi}{N}\right) \right]_{j=1,\dots,N}$$ autovettori.

Consideriamo l'errore iniziale e_0, possiamo scriverlo come comb. lin degli autovettori (che sono una base)

$$\rightsquigarrow e_0 = \sum_{i=1}^{N-1} c_i w^{(i)}$$

Da cui otteniamo: $e_k = B_{J,\omega}^k e_0 = \sum_{i=1}^{N-1} c_i B_{J,\omega}^k w^{(i)} = \sum_{i=1}^{N-1} c_i \lambda_i^k(B_{J,\omega}) w^{(i)}$

Consideriamo ora l'andamento degli autovalori al variare di ω:



Abbiamo che per $\omega \in (0,1)$

$$|\lambda_i(B_{J,\omega})| < 1 \qquad \forall i$$

$$\lambda_1(B_{J,\omega}) = 1 - 2\omega \operatorname{sen}^2\left(\frac{\pi}{2N}\right) \underset{N=1/h}{=} 1 - 2\omega \operatorname{sen}^2\left(\frac{\pi h}{2}\right) = 1 - O(h^2)$$

più grande

Quindi l'autovalore più grande rimane sempre vicino a 1, indipendentemente da ω! ω non ha nessun ruolo da giocare su λ_1.

Inoltre osserviamo che al tendere di h a zero $\lambda_1 \to 1$ e quindi il metodo rallenta.

L'osservazione chiave è che ω può giocare un ruolo in alcune frequenze mentre altre non le tocca.

Per come sono fatti gli autovalori ho che i piccolo corrisponde alle basse frequenze e i grande alle alte frequenze.

Ho due per le basse frequenze ω non va mai bene, e non ci posso fare nulla, mentre per le alte frequenze ω può giacere un ruolo, perché vediamo due per esempio per $\omega = 1/2$ gli ultimi autovalori vanno a zero e li possiamo "eliminare".
(Nella pratica si sceglie $\omega = 2/3$ perché così ho $|\lambda_{N/2}| = |\lambda_N|$ e invece di sbattersi a zero so che li limito entrambi e ovviamente anche quelli compresi).

def: **SMOOTHING FACTOR**: $S = \max_{\frac{N}{2} \le i \le N} |\lambda_i(B_{J,\omega})|$.

Ho due per $\omega = 2/3 \rightarrow S = 1/3$ $\quad |\lambda_{N/2}| = |\lambda_N| = 1/3$ $\quad$ e $|\lambda_i| < 1/3$ $\quad N/2 < i < N$.

OSS: Anche Gauss-Seidel e gli altri metodi stazionari si comportano analogamente, ovvero due non riescono a smorzare le basse frequenze ma riescono a sumissare le alte.

Ora dobbiamo pensare di "spezzare il problema" in più spazi (visto che ho come diviso il mio problema tra basse e alte frequenze e riesco a sistemare le cose da una parte sì e una no)

Al posto di considerare $Ax = b$ passiamo all'equazione del residuo

$$Ae = r.$$

Osserviamo che i due sistemi sono equivalenti infatti: sia $x^* = A^{-1}b$

$e = x^* - \tilde{x}$ con $\tilde{x}$ approssimazione di x^*

$Ae = A(x^* - \tilde{x}) = b - A\tilde{x} = r$

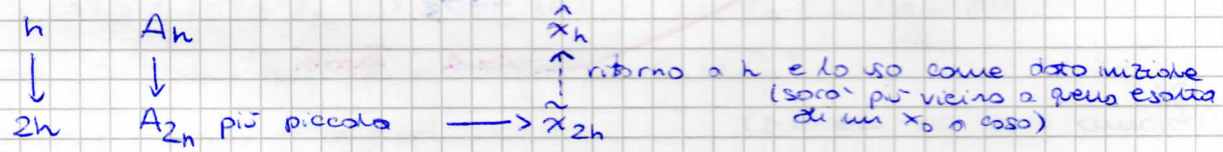
Poi si recupera x^* facendo $x^* = \tilde{x} + e$.

Ora l'idea è di "confondere" il metodo per sistemare anche le basse frequenze.

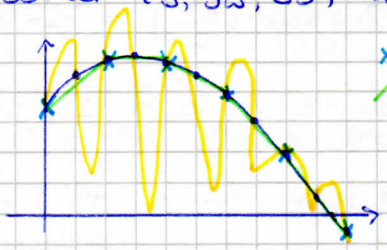
Proiettiamo il problema su una mesh più rada (COARSE GRID), così abbiamo anche:

- minor costo computazionale
- velocità di convergenza leggermente maggiore (tipo se $h \rightarrow 2h$ ho $1 - O(h^2) \rightarrow 1 - O(4h^2)$)

Vediamo un'idea di cosa possiamo fare:



Punto chiave: $Ae = r$.
Dopo smoother (J, J_ω, GS) l'errore è smooth (bassa frequenza)

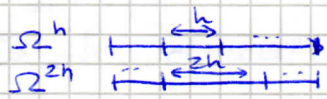


UNA FUNZIONE SMOOTH VIENE APPROSSIMATA BENE CON L'INTERPOLAZIONE LINEARE!
(se non è smooth no!)

Quindi se proietto il problema su una griglia più grande non perdo molto.

Un'altra cosa bella è che il fatto di essere alta o bassa frequenza dipende dalla mesh! Ovvero una cosa che magari è in media frequenza su una mesh più fitta può diventare in alta frequenza su una mesh più rada! (perché le frequenze sono indipendenti dalla mesh, è la loro "classificazione" che cambia.)

IDEA QUALITATIVA:



SMOOTHER • Risolvo $Ax=b$ su Ω^h
(applico lo smoother $B_{JW}, B_{GS}, \dots$)
$\rightsquigarrow$ trovo $\tilde{x}_h$ (cioè non è che lo risolvo ma gli faccio fare un po' di giri)

EQ RESIDUO • Calcolo $r = b - A\tilde{x}_h$

COARSEGRID CORRECTION • "Proietto" su Ω^{2h} COARSE GRID e risolvo $A_{2h}\tilde{e}_{2h} = r_{2h}$.
Poi "prolungo" $\tilde{e}_{2h}$ a un $\tilde{e}_h$ (per tornare alla griglia Ω^h)

• $x_h = \tilde{x}_h + \tilde{e}_h$

Dobbiamo trovare un modo per passare da Ω^h a Ω^{2h} e viceversa.
(OPERATORE di GRID TRANSFER)

Chiamerò: RESTRITTORE $r_h \longrightarrow r_{2h}$
PROLUNGATORE $\tilde{e}_{2h} \longrightarrow \tilde{e}_h$.

N.B. r_{2h} e $\tilde{e}_h$ non sono il residuo e l'errore "esatti" perché sono calcolati col restrittore ecc..

Inoltre dovremo considerare cos'è A_{2h} (vedremo che come è intuitivo è la "restrizione" di A_h alla mesh più rada).

* Vediamo prima come definiamo il PROLUNGATORE: $I_{2h}^h : \Omega_{2h} \longrightarrow \Omega_h$ (coarse → fine)

$I_{2h}^h v^{2h} = v^h$ t.c.
$$\begin{cases} (v^h)_{2i} = (v^{2h})_i & i = 0, \dots, N_2 \\ (v^h)_{2i+1} = \frac{1}{2}\left[(v^{2h})_i + (v^{2h})_{i+1}\right] & i = 0, \dots, N_2 - 1 \end{cases}$$

la matrice della trasformazione è:

$$C = \begin{bmatrix} \frac{1}{2} & & & & & \\ 1 & & & & & \\ \frac{1}{2} & \frac{1}{2} & & & & \\ & 1 & & & & \\ & \frac{1}{2} & \frac{1}{2} & & & \\ & & 1 & & & \\ & & \frac{1}{2} & \frac{1}{2} & & \\ & & & 1 & & \\ & & & \frac{1}{2} & & \end{bmatrix}$$

(7×3 nell'es del disegno)

* RESTRITTORE (1): $I_h^{2h} : \Omega^h \longrightarrow \Omega^{2h}$ (fine → coarse) (iniezione)

$I_h^{2h} v_h = v_{2h}$ t.c. $(v^{2h})_i = (v^h)_{2i}$ $i = 0, \dots, N_2$

(Nel disegno sopra è come se parto dalla verde e poi tengo solo i pallini blu)
Quindi tengo soltanto le informazioni che mi servono.

La matrice è $\begin{bmatrix} 0 & 1 & & & & & \\ & & 0 & 1 & & & \\ & & & & 0 & 1 & 0 \end{bmatrix}$ (3×7).

In realtà con questo restrittore la A_{2h} viene brutta quindi se ne usa un altro:

* RESTRITTORE (2): $I_h^{2h} v_h = v_{2h}$ t.c. $(v^{2h})_i = \frac{1}{4}\left[(v^h)_{j-1} + 2(v^h)_j + (v^h_{j+1})\right]$ FULL-WEIGHTING

La matrice è $\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & \\ & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & \\ & & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} = \frac{1}{2} C^T$

↳ cosa che ci piacerà molto

SCHEMA di un PASSO di METODO:

$x^h_{new} = \text{Metodo}(x^h_{old}, b^h, A^h, \alpha_1, \alpha_2)$

→ tipicamente $\alpha_i = 1, 2, 3$ non di più

1) Rilasso α_1 volte $A_h u_h = b_h$ in Ω^h con innesco arbitrario x^h_{old} (PRESMOOTHER) $\rightsquigarrow \tilde{x}_h$

2) Calcolo $r_h = b_h - A_h \tilde{x}_h$

3) Restrittore: $\tilde{r}_{2h} = I_h^{2h} r_h$ proiezione del residuo (non è quello vero) in Ω^{2h}

4) "Risolvo" $A_{2h} \tilde{e}_{2h} = \tilde{r}_{2h}$

5) Prolungatore $\tilde{e}_h = I_{2h}^h \tilde{e}_{2h}$

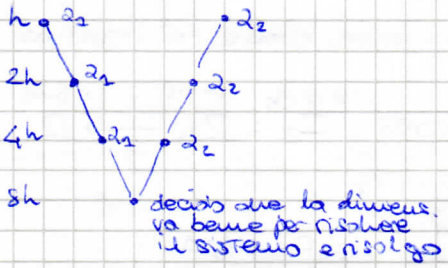
6) Correggo approssimazione smoother: $\hat{x}_h = \tilde{x}_h + \tilde{e}_h$

7) Rilasso α_2 volte $A_h u_h = b_h$ con innesco $\hat{x}_h$ (POST SMOOTHER) $\rightsquigarrow x^h_{new}$ (nel caso si fossero reintrodotte alte frequenze a causa degli errori nel non considerare il residuo esatto)

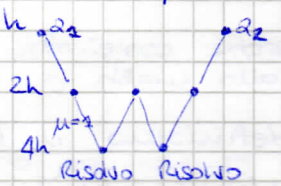
Questo metodo proposto così è detto TWO-GRID, se al punto (4) invece di risolvere perché la dim è troppo grande riapplico la ricorsione aggiungo una griglia...

Abbiamo più metodi per "approssimare":

V-cycle



W-cycle



15/01/2013

PROPRIETÀ di CONVERGENZA

$\{\tau_k\}_{k \geq 1}$ mesh t.c. $\tau_{k-1} \subset \tau_k \rightarrow V_{k-1} \subset V_k \quad \forall k$ (τ_1 mesh + coarse)

↕ cioè la mesh successiva si ottiene per raffinamento. → ✗ no △ sì

Inoltre si suppone che la mesh non sia troppo brutta ("quasi uniforme", "regolare")

TEO: (di CONVERGENZA per il TWO GRID)

$\exists C > 0$ indipendente da k (indice di mesh) t.c.

$$\| \hat{e}_{k+1} \|_E \leq C \mu^{-1/2} \| e_0 \|_E$$

dove abbiamo indicato con $\| \ \|_E = (a(\ ,\ ))^{1/2}$ (norma energia) e

$\hat{e}_{k+1} = z - TG(z_0)$ (innesco), μ = # passi di smoother

z = sol. esatta FEM $Az = b$

Il teorema in pratica dice che la TG è una contrazione con fattore di contrazione indipendente da k.

TEO: (di CONVERGENZA per V-CYCLE a K livelli)

Sia m = # passi di Smoother, vale:

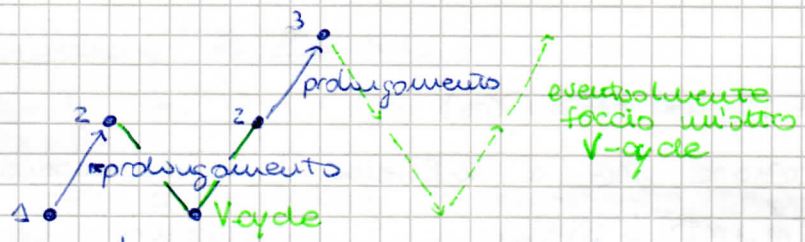
$$\| z - MG_m(K, z_0) \|_E \leq \frac{C^*}{C^* + m} \| z - z_0 \|_E$$

($MG_m(K, z_0)$: innesco; soluzione esatta FEM $Az = b$; livello di multigrid; approssimazione dato dal multigrid; $\frac{C^*}{C^*+m} = \gamma$)

Dunque $\forall m$ MG è una contrazione e γ è indipendente da K (= # livelli di ricorsione)

FULL MULTIGRID:

IDEA: Cercare l'innesco migliore ⇝ così ottengo velocità convergenza maggiore



$u_1 = A_1^{-1} b_1$ soluzione "esatta" (sulla mesh coarse)

ALGORITMO:

$k=1$ (più coarse) $\quad \hat{u}_1 = A_1^{-1} b_1$ sol esatta (↭ metodo diretto)

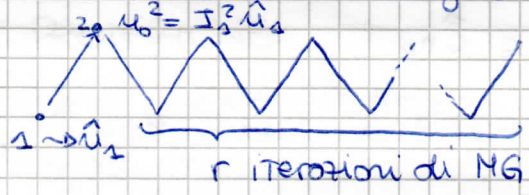
$k \geq 2 \quad u_0^k = I_{k-1}^k \hat{u}_{k-1}$ innesco

$u_\ell^k = MG(k, u_{\ell-1}^k, b_k) \quad 1 \leq \ell \leq r$ [nel disegno ho preso $r=1$]

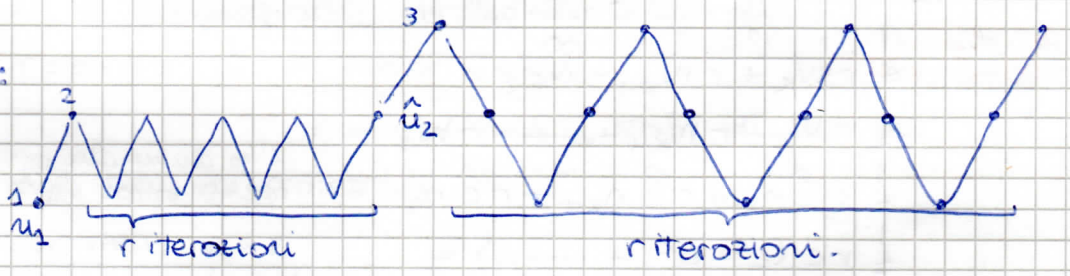
$\hat{u}_k = u_r^k$

Facciamo di nuovo un disegno per capire l'algoritmo

$k=2$:



$k=3$:



Ora vediamo un risultato che regge a legare la soluzione approssimata alla soluzione nel continuo. (caso molto bello, di solito leghiamo l'approssimata alla soluzione esatta nel discreto)

u = soluzione esatta del problema variazionale

u_k e V_k approssimazione a FE lineari di u.

Supponiamo (vedi Russo) che vale $\|u - u_k\|_{H^s(\Omega)} \leq C h_k^{r-s} \|u\|_{H^r(\Omega)}$ $\quad s = 0, 1$ (h_k: max diametro di $T \in \mathcal{T}_k$)

TEO: Se l'iterazione MG a k livelli è una contrazione con fattore indipendente da k allora $\exists C > 0$ t.c.

$$\|u_k - \hat{u}_k\|_E \leq C h_k \|u\|_{H^2(\Omega)}$$

(u_k: sol. app. FEM; $\hat{u}_k$: approx. FULL MULTIGRID di u_k)

(è un risultato analogo a quello sopra ☺)

COSTO COMPUTAZIONALE (LINEARE)

Per prima cosa vogliamo dare una stima asintotica di dim $V_k = n_k$

Per la formula di Eulero Abbiamo: $V_k^I - e_k^I + t_k = 1$ (vertici interni, edge int., triangoli $\mathcal{T}_k$)

$$\Rightarrow n_k = V_k^I = 1 + e_k^I - t_k$$

Abbiamo due: $\begin{cases} e_k^I = 2 e_{k-1}^I + 3 t_{k-1} \\ t_k = 4 t_{k-1} \end{cases}$

da cui si ottiene:

$$n_k \sim \frac{t_1}{8} \cdot 4^k$$

PROP: le full multigrid ($p<4$) ha costo $O(n_k)$

dim: • COSTO dell'iterazione MG a k-livelli? Lo chiamiamo W_k

$$W_k \leq \underbrace{C n_k (m_1 + m_2)}_{\text{costo prodotto matrice vettore MENO: A è sparsa}} + \underbrace{p}_{\text{costo delle iterazioni}} W_{k-1}$$

(m_1 = # iter presmoother, m_2 = # it post)

p = # di chiamate: $p=1$ Vcycle, $p=2$ Wcycle

$$\leq C \tilde{m} n_k + p(C \tilde{m} n_{k-1} + p W_{k-2}) =$$
$$\leq C \tilde{m} n_k + p C \tilde{m} n_{k-1} + \dots + p^{k-1} C \tilde{m} n_1 =$$
$$\leq \hat{c} \tilde{m} 4^k + p \hat{c} \tilde{m} 4^{k-1} + \dots + p^{k-1} \hat{c} \tilde{m} 4 = \quad \text{con } \hat{c} = \frac{C t_1}{8}$$
$$\leq \hat{c} \tilde{m} 4^k \left(1 + \frac{p}{4} + \dots + \frac{p^{k-1}}{4^{k-1}}\right) =$$
se $\frac{p}{4} < 1$ $\leq \hat{c} \tilde{m} 4^k \frac{1}{1 - p/4} \leq \hat{\hat{c}} n_k$

• $\hat{W}_k$ = costo full multigrid a livello k

$$\hat{W}_k = \hat{W}_{k-1} + r W_k =$$
(r = # iterazioni di multigrid)
$$\leq r W_k + r W_{k-1} + \hat{W}_{k-2} =$$
$$\leq r W_k + r W_{k-1} + \dots + r W_1 =$$
$$\leq r \hat{c} n_k + r \hat{c} n_{k-1} + r \hat{c} n_1 =$$
$$\leq r c (4^k + 4^{k-1} + \dots + 4) =$$
$$= r c 4^k \left(1 + \frac{1}{4} + \dots + \frac{1}{4^{k-1}}\right) \leq r C 4^k \frac{1}{1 - 1/4} \leq C n_k \quad \blacksquare$$

l'ho chiamata così perché è quella sopra, poi la chiamo sempre C

Ora torniamo indietro e cerchiamo di vedere nuovamente il metodo multigrid come metodo iterativo stazionario.
Esaminiamo il TWOGRID per semplicità. [presmoother + coarse grid correction]

$$\tilde{x}^h = G_h^{pre} x_{old}^h + c$$

CGC
$$\begin{cases} r^h = b^h - A \tilde{x}^h \\ \tilde{r}^{2h} = I_h^{2h} r_h \\ A_{2h} \tilde{e}_{2h} = \tilde{r}_{2h} \\ \tilde{e}_h = I_{2h}^h \tilde{e}_{2h} \\ x_{new}^h = \tilde{x}^h - \tilde{e}^h \end{cases}$$

$$x_{new}^h = \tilde{x}^h + I_{2h}^h \tilde{e}_{2h} = \tilde{x}^h + I_{2h}^h A_{2h}^{-1} r_{2h} = \dots = (\underbrace{I_h}_{\text{identità}} - I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h) \tilde{x}^h =$$
$$= \underbrace{(I_h - I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h)}_{C_h \text{ coarse grid correction}} G_h^{pre} x_{old}^h + \dots$$

se uso postsmoother compare G_h^{post} qui davanti

$\Rightarrow$ METODO ITERATIVO STAZIONARIO

OSS: Dato che è un metodo iterativo stazionario la decrescita dell'errore può essere al più lineare (quindi non può comportarsi come il CG in presenza di cluster - ad esempio -)

def: P **PROIETTORE (OBLIQUO)** se $P^2 = P$

OSS: se P proiettore $\Rightarrow \Sigma_P \subseteq \{0,1\}$ (spettro)

dim: $Px = \lambda x \Rightarrow P^2x = \lambda^2 x \Rightarrow \lambda^2 - \lambda = 0 \Rightarrow \lambda(\lambda-1) = 0$ ∎
$P^2x = Px$

Dunque $\rho(P) = 1$ e quindi P non è un metodo iterativo stazionario convergente.

PROP: $C_h = I_h - I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h$ è un proiettore.

dim: $C_h^2 = (I_h - I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h)(I_h - I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h) =$ $= A_{2h}$
$\overset{!}{=} (I_h - 2 I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h + I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h) =$ (sono uguali)
$\overset{!}{=} I_h - I_{2h}^h A_{2h}^{-1} I_h^{2h} A_h = C_h$ ∎

Si può però dimostrare che C_h è un proiettore ortogonale se si usa il prodotto scalare $(\,,)_A$ (quello del CG) [Questo è un legame con CG ma non ho capito perché]

16/01/2013 (20)

METODI di PROIEZIONE

Vogliamo risolvere il sistema $Ax = b$. L'idea è quella di cercare una soluzione approssimante in un certo sottospazio $K \subseteq \mathbb{R}^n$, $\dim K = m$. (ossia soddisfacente determinati vincoli).

Tipicamente si scelgono dei vincoli di ortogonalità:

$r = b - Ax \perp$ m vettori l.i. — CONDIZIONI di PETROV-GALERKIN

m vincoli $\Rightarrow$ $\mathcal{L}$ STSP $\dim \mathcal{L} = m$

Ci sono due tipi di proiettori:

- ORTOGONALI $\mathcal{L} = K$
- OBLIQUI $\mathcal{L} \neq K$

Il problema è quindi:

Trovare $\tilde{x} \in K$ sol. appross. t.c. $b - A\tilde{x} \perp \mathcal{L}$

che introducendo il vettore di innesco x_0 possiamo riscrivere:

Trovare $\tilde{x} \in x_0 + K$ t.c. $b - A\tilde{x} \perp \mathcal{L}$

→ Metodi di proiezione in STSP di Krylov.

def: $K_m(M, v)$ $M \in \mathbb{C}^{n\times n}$ $v \in \mathbb{C}^n$, $K_m(M,v) := \langle v, Mv, M^2v, \dots, M^{m-1}v \rangle$
$\overset{!}{=} \{u \in \mathbb{C}^n$ t.c. $u = p_{m-1}(M)v\}$ (p_{m-1} = polinomio di grado $m-1$)

def: **METODO di PROIEZIONE in STSP di KRYLOV** per risolvere $Ax = b$. Cerco x_k soluzione approssimante in $x_0 + K_k$ con x_0 vettore di innesco arbitrario, con $K_k = K_k(A, r_0) = \langle r_0, Ar_0, \dots, A^{k-1}r_0 \rangle$
e inoltre si impone $r_k = b - Ax_k \perp \mathcal{L}_k$ STSP t.c. $\dim \mathcal{L}_k = k$.

Abbiamo ancora diverse possibilità di scelta perché dobbiamo fissare $\mathcal{L}$.
$\mathcal{L}_k = K_k(A, r_0)$ o $\mathcal{L}_k = AK_k(A, r_0)$ (GMRES)
$\mathcal{L}_k = K_k(A^T, r_0)$

1° PASSO: Costruire una base ortonormale $K_k(A, r_0)$.

se abbiamo $v_1 \to v_k$ vettori linearmente indipendenti possiamo ortonormalizzarli col metodo di GRAM-SCHMIDT (classico).

Metodo classico **GRAM-SCHMIDT**:

$v_1 \to v_n$ l.i.

$$\tilde{u}_k = v_k - \sum_{i=1}^{k-1} \langle v_k, u_i \rangle u_i$$

$$u_k = \tilde{u}_k / \|\tilde{u}_k\|_2$$

Metodo **GM modificato** (per l'aritmetica floating point)

$u_1 = v_1 / \|v_1\|_2$

per $k = 2 \to n$

$\tilde{u}_k = v_k$

per $i = 1 \to k-1$ ← procedo a toppe rispetto al vettore già modificato. (È meglio in aritmetica floating point perché è come se avessi $\tilde{u}$ più "pulito")

$\tilde{u}_k = \tilde{u}_k - \langle \tilde{u}_k, u_i \rangle u_i$

$u_k = \tilde{u}_k / \|\tilde{u}_k\|_2$

Metodo di **ARNOLDI** = Metodo GM modificato per costruire basi ortonormali per $K_k(A, q_1)$

INPUT: ? A, q_1 ?

OUTPUT: $q_1 \to q_k$ vettori ortonormali e base per $K_k(A, q_1)$

q_1 t.c. $\|q_1\|_2 = 1$ (ottenuto lo normalizzo)

per $j = 1 \to k$

$\tilde{q}_{j+1} = A q_j$

per $i = 1 \to j$ ← è giusto perché è già avanti di un passo

$h_{ij} = \langle \tilde{q}_{j+1}, q_j \rangle$

$\tilde{q}_{j+1} = \tilde{q}_{j+1} - h_{ij} q_i$

$h_{j+1,j} = \|\tilde{q}_{j+1}\|_2$

$q_{j+1} = \tilde{q}_{j+1} / h_{j+1,j}$ (•)

OSS: Non è ovvio che quella che abbiamo trovato sia una base per $K_k(A, q_1)$ (è ovvio che sono ortonormali ma non che sia base)

PROP 1: $\{q_1 \to q_k\}$ dell'algoritmo di ARNOLDI formano una base ortonormale per $K_k(A, q_1)$

dim: Sono ortonormali per costruzione. Vogliamo verificare che

$$q_j = P_{j-1}(A)\, q_1$$

basta che scelgo: $P_0(t) \equiv 1$

Per induzione: $j = 1$ vero $q_1 = p_0(A) q_1$

hp di induzione $i \le j$ $q_i = p_{i-1}(A) q_1$

Verifichiamo per $i = j+1$. Per def dell'algoritmo abbiamo:

$$h_{j+1,j}\, q_{j+1} \overset{1}{=} \tilde{q}_{j+1} = A q_j - \sum_{i=1}^{j} h_{ij} q_i =$$

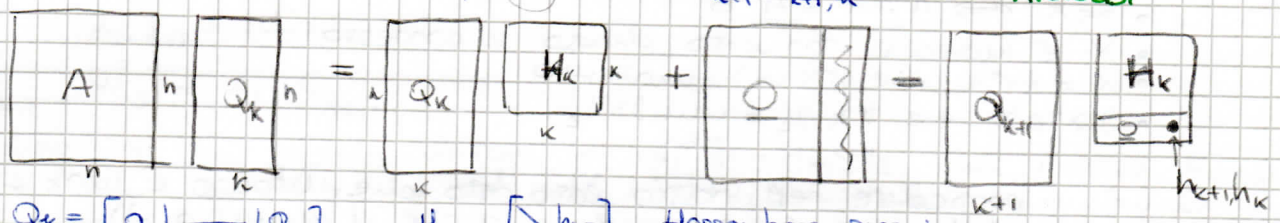
$$\pm A p_{j-1}(A) q_1 - \sum_{i=1}^{j} u_{ij} p_{i-1}(A) q_1 = \tilde{p}_j(A) q_1$$ ■

PROP 2: L'algoritmo di Arnoldi si può riscrivere come:

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T = Q_{k+1} H_{k+1,k}$$

(vettore)

HOUSEHOLDER - ARNOLDI



dove $Q_k = [q_1 | \cdots | q_k]$ $\quad H_k = \begin{bmatrix} h_{ij} \\ 0 \end{bmatrix}$ Hessenberg superiore

Infatti: $j=1$ per $i=1$ $\quad \tilde{q}_2 = Aq_1 - h_{11} q_1$

$$\rightarrow Aq_1 = h_{11} q_1 + \tilde{q}_2 \underset{(\bullet)}{=} h_{11} q_1 + h_{21} q_2$$

$j=2$ per $i=1$ $\quad \tilde{q}_3 = Aq_2 - h_{12} q_1$ (vedi l'algoritmo per verificare le uguaglianze - che sono giuste, controllare →)

$i=2$ $\quad \tilde{q}_3 = \tilde{\tilde{q}}_3 - h_{22} q_2 = Aq_2 - h_{12} q_1 - h_{22} q_2$

$$\rightarrow Aq_2 = h_{12} q_1 + h_{22} q_2 + \tilde{q}_3 \underset{(\bullet)}{=} h_{12} q_1 + h_{22} q_2 + h_{32} q_3$$

Quindi in generale ricaviamo:

$$Aq_k = h_{1k} q_1 + h_{2k} q_2 + \cdots + h_{kk} q_k + h_{k+1,k} q_{k+1}$$ ■

GMRES (matrici non Hermitiane)

È un metodo di proiezione su:

$$K_k(A, q_1) = \langle q_1, Aq_1, \cdots, A^{k-1} q_1 \rangle \qquad q_1 = \frac{r_0}{\|r_0\|_2}$$

$$\mathcal{L}_k = A K_k(A, q_1)$$

Cerco $x_k = x_0 + K_k(A, q_1)$

ne ho una base ortonormale $q_1, \cdots, q_k$ → NB è un modo per scrivere una comb. lineare dei vettori q_i (con y_i come coefficienti)

$$= x_0 + Q_k y$$

Il mio obiettivo è minimizzare la norma 2 del residuo: $\|r_k\|_2$

$$r_k = b - Ax_k = b - A(x_0 + Q_k y) = r_0 - AQ_k y$$

Posto $\beta = \|r_0\|_2$ $\quad e_1 = [1, 0, \cdots, 0]^T \in \mathbb{C}^{k+1}$ per costruzione di Arnoldi abbiamo:

$$Q_{k+1} e_1 = q_1 := \frac{r_0}{\beta}$$

E quindi abbiamo $r_0 = \beta Q_{k+1} e_1$. (★)

Pb: Trovare $\min_{y \in \mathbb{C}^k} \|r_k\|_2$. Abbiamo

$$\min_{y \in \mathbb{C}^k} \|r_k\|_2 = \min_{y \in \mathbb{C}^k} \|r_0 - AQ_k y\| =$$

$$= \min \|\beta Q_{k+1} e_1 - Q_{k+1} H_{k+1,k} y\|_2$$

(★)+PROP

$$= \min \|Q_{k+1} (\beta e_1 - H_{k+1,k} y)\|_2 \qquad \text{MEMO: } Q_{k+1}^H Q_{k+1} = I$$

$$\underset{(■)}{=} \min_{y \in \mathbb{C}^k} \|\beta e_1 - H_{k+1,k} y\|_2$$

La matrice ha rango massimo per costruzione
⇒ Pb lineare di minimi quadrati ha un'unica soluzione.

TEO (MENO): Dato il problema ai minimi quadrati lineari:

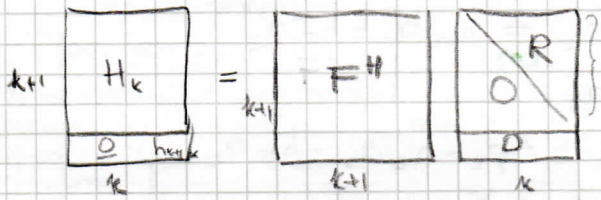
"Determinare $X = \{x \in \mathbb{C}^m \mid \|Ax-b\|_2 = \min_{y \in \mathbb{C}^n} \|Ay-b\|\}$ $A \in \mathbb{C}^{m \times n}$ y

Vale:

- $x \in X \iff A^H A x = A^H b$
- X è insieme non vuoto, chiuso e convesso
- $X = \{x^* y\}$ sse $rk(A)$ massimo
- $\exists\, x^* \in X$ t.c. $\|x^*\|_2 = \min_{x \in X} \|x\|_2$.

Quindi in particolare nel nostro caso dato che il rango è max abbiamo un'unica soluzione (che quindi è anche il minimo 2).

Sia $H_{k+1,k} = F^H R$ fattorizzazione QR di una matrice rettangolare (F sarebbe la Q ma ho già troppe Q in giro)



Dunque:

$$(\blacksquare) = \min_{y \in \mathbb{C}^k} \|\beta e_1 - F^H R y\|_2 =$$

$$= \min_{y \in \mathbb{C}^k} \|F^H(\beta F e_1 - R y)\|_2 =$$

(vero siccome l'ultima riga è nulla considero solo le prime k componenti, e minimizzo su quelle, dove è agevole.)

$$\overset{!}{=} \min_{y \in \mathbb{C}^k} \|\beta F e_1 - R y\|_2 = \min_{y \in \mathbb{C}^k} \|(\beta F e_1)_{k+1} - R_{kk}\, y_{k+1}\|$$

Ora abbiamo ottenuto un sistema quadrato con $R_{k \times k}$ non singolare! Non dobbiamo più risolvere un problema di minimizzazione!

$\Rightarrow y \in \mathbb{C}^k$ cercato è la soluzione di:

$$R_{kk}\, y = \beta (F e_1)_{k+1}$$

e vale inoltre $\|r_k\|_2 = \|b - A x_k\|_2 = \|\beta F e_1 - R y\|_2 = \beta (F e_1)_{k+1}$ (mi rimane solo l'ultima componente perché sulle prime k $\equiv$)

OSS: Questo metodo è abbastanza costoso! Ma d'altronde i sistemi non hermitiani sono cose brutte (non sono miei pensieri domestici :-)
Costoso soprattutto in termini di memoria.

Ci sono altre versioni del GMRES oltre a quello "classico":

- con RESTART
- con TRONCAMENTO

} derivano dal problema del costo di memoria

Idea restart: Faccio m iterazioni $q_1, \dots, q_m$ mi fermo e prendo l'x_m trovato come nuovo innesco.

Idea troncamento: Non faccio l'ortonormalizzazione completa, cioè non torno indietro fino all'indice 1. Faccio una "quasi-ortonormalizzazione".

(21) 18/01/2013

ALGORITMO: (per GMRES) [Saad p 158]

1. Calcolo $r_0 = b - A x_0$, $\beta = \|r_0\|_2$, $v_1 = r_0/\beta$
2. Dimensionare (allocare la memoria) $H_m = [h_{ij}]_{i=1,\dots,m+1;\, j=1,\dots,m} = 0$
3. per $j = 1 \to m$

   (ORTOGONALIZZAZIONE)

   Calcolo $w_j = A v_j$

N.B. Ho scritto per forza maiore che usiamo tanta memoria

per $i = 1 \to j$

$h_{ij} = (w_j, v_i) = w_j^T v_i$

$w_j = w_j - h_{ij} v_i$

$h_{j+1,j} = \|w_j\|_2$

$v_{j+1} = w_j / h_{j+1,j}$

4. Calcolare y_m minimizzatore di $\|\beta e_1 - \hat{H}_m y\|_2$ [QR e risolvo "$Ry = b$"]
e $x_m = x_0 + V_m y_m$ ove $V_m = [v_1 | v_2 | \dots | v_m]$.

ANALISI PROPRIETA' di CONVERGENZA (GMRES)

Al passo k ho: $x_k = x_0 + z_k$ con $z_k \in K_k(A, r_0) = \langle r_0, Ar_0, \dots, A^{k-1} r_0 \rangle$

$r_k = b - Ax_k = b - Ax_0 - Az_k = r_0 - Az_k \in \langle r_0, Ar_0, \dots, A^k r_0 \rangle$

Ossia: $r_k = p_k(A) r_0$ con p_k polinomio di grado k t.c. $p_k(0) = 1$

$\stackrel{!}{=} r_0 + \sum_{i=1}^{k} \gamma_i A^i r_0$

Dunque possiamo scrivere la norma dell'errore (lo minimizza perché? boh)

$$\|r_k\|_2 = \min_{\substack{p_k \in \mathbb{P}_k \\ p_k(0)=1}} \|p_k(A) r_0\|_2$$

Ora vogliamo trovare una maggiorazione per il residuo.

PROP: Se $A = V \Lambda V^{-1}$ con V matrice con colonne gli autovettori (normalizzati in modo opportuno)

NEW ENTRY!

$$\Rightarrow \frac{\|r_k\|_2}{\|r_0\|_2} \le K_2(V) \cdot \min_{\substack{p_k \in \mathbb{P}_k \\ p_k(0)=1}} \left\{ \max_{i=1,\dots,n} |p_k(\lambda_i(A))| \right\}$$

OSS: Se A è normale $\Rightarrow V$ unitaria e dunque $\|V\|_2 = \|V^{-1}\|_2 = 1$
$\Rightarrow K_2(V) = 1$ e il termine "nuovo" sparisce.

Se A non è normale in generale $K_2(V) \neq 1$ e ce lo dobbiamo tenere!

dim: $\|r_k\|_2 = \min_{\substack{p_k \in \mathbb{P}_k \\ p_k(0)=1}} \|p_k(A) r_0\|_2 = \min \|p_k(V \Lambda V^{-1}) r_0\|_2 =$

dato che $A^s = V \Lambda^s V^{-1}$ ←

$= \min \|V p_k(\Lambda) V^{-1} r_0\|_2 \le \min \|V\|_2 \, \|p_k(\Lambda)\|_2 \, \|V^{-1}\|_2 \, \|r_0\|_2 =$

$\stackrel{!}{=} \min \|p_k(\Lambda)\|_2 \, \|r_0\|_2 \cdot K_2(V)$

$\stackrel{\text{def di } \|\cdot\|_2}{\stackrel{!}{=}} K_2(V) \min_{\substack{p_k \in \mathbb{P}_k \\ p_k(0)=1}} \left\{ \max_{i=1,\dots,n} |p_k(\lambda_i(A))| \right\} \|r_0\|_2$ ■

OSS$_1$: Non è noto se questa è una stima stretta! (diversamente che per il GC), questo perché non è detto che il polinomio che minimizza con dentro r_0 sia lo stesso che minimizza senza r_0.

Se A è normale però ho $K_2(V) = 1$ e dunque la limitazione è stretta.

OSS$_2$: Anche questo metodo è sensibile alla presenza di cluster. Se i cluster sono lontani dall'origine si comporta bene, se sono intorno all'origine no. ☺

OSS$_3$: Se A non è normale, ma V è "abbastanza" ben condizionata allora la PROP ci dà una stima non stretta ma ragionevole. Inoltre sono sostanzialmente solo gli autovalori a determinare le proprietà di convergenza (il ruolo degli autovettori è marginale).

Un altro strumento che viene usato per studiare (le proprietà) la velocità di convergenza è il seguente:

Si consideri $F(A) = \{ y^H A y \mid y \in \mathbb{C}^n,\ y^H y = 1 \}$ (FIELD of VALUES), $A \in \mathbb{C}^{n \times n}$

$$= \left\{ \frac{y^H A y}{y^H y} \mid y \in \mathbb{C}^n, y \neq 0 \right\}$$

Si noto che questo coso "assomiglia" ai quozienti di Rayley. Quindi l'idea è quello di partire da quello che sappiamo nel caso simmetrico e cercare di estenderlo al caso non simmetrico.

PROBLEMI di CONVEZIONE - DIFFUSIONE — FEM

(Streamline diffusion, SUPG = streamline upwind Petrov-Galerkin)

$$\begin{cases} -\operatorname{div}(\varepsilon \nabla u) + \beta \nabla u = f & \text{su } \Omega \\ u = 0 & \text{su } \partial\Omega \end{cases}$$

FEM: $a(u,v) = F(v) \quad \forall v \in V$

$$a(u,v) = \int_\Omega \varepsilon \nabla u \nabla v + \int_\Omega \beta \cdot \nabla u\, v$$

$$F(v) = \int_\Omega f v$$

def: Pe NUMERO di PECLET che misura rapporto tra diffusione e convezione.

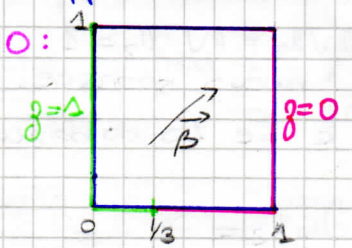
Si può mostrare che se $Pe > 1$ localmente, FEM non funziona. Detto meglio

Possiamo definire anche un numero di Peclet locale:

$$Pe_K = \frac{\|\vec{\beta}_K\| |h_K|}{2\varepsilon_K}$$ h_K diam K con $K \in \mathcal{T}$ triangolo.

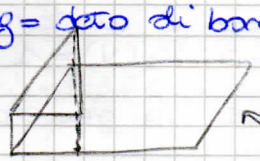
Se $Pe_K > 1$ FEM non funziona, ma proprio non funziona il metodo dell'approssimazione con gli elementi finiti!

ESEMPIO:



$$\vec{\beta} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

(g = dato di bordo)



← c'è un salto nelle condizioni di bordo.

Noi ora fisseremo h_K e $\vec{\beta}$ e vedremo cosa succede facendo tendere $\varepsilon \to 0$. (negli esempi, credo)

Vediamo un schema del metodo SUPG. Consideriamo il problema (già proiettato sullo spazio più piccolo triangoli))

$$\int_\Omega \varepsilon \nabla u_L \nabla v_L + \int_\Omega (\vec{\beta} \nabla u_L) v_L + \boxed{\sum_{K \in \mathcal{T}} \tau_K \int_K (\vec{\beta}_K \nabla u_L)(\vec{\beta}_K \nabla v_L)} =$$

$$= \int_\Omega f v_L + \boxed{\sum_{K \in \mathcal{T}} \tau_K \int_K f_K (\vec{\beta}_K \nabla v_L)}$$

→ ho correzione del termine noto e anche il (termine a sx) coef. della matrice

dove abbiamo posto:

$$\tau_K = \frac{h_K^{\beta_K}}{3\|\vec{\beta}_K\|}$$ con $h_K^{\beta_K}$ è il "diametro nella direzione di $\vec{\beta}$"

se $Pe_K > 1$ e altrimenti $\tau_K = 0$ (se $Pe_K < 1$).

Vediamo come possiamo approssimare questi nuovi integrali che abbiamo introdotto.

formula del baricentro

$$\tau_K \int_K \beta \nabla\varphi_j \nabla\varphi_i \cong \frac{\tau_K \; dot(\vec{\beta}_K(b_K) e_j^{\perp}) \cdot dot(\vec{\beta}_K(b_K) e_i^{\perp})}{4|T|}$$

$$\tau_K \int_K f_K (\vec{\beta}_K \cdot \nabla\varphi_i) \cong \tau_K \frac{f_K(b_K)}{2} \; dot(\vec{\beta}_K(b_K), e_i^{\perp})$$

ESEMPIO: (da provare con matlab) $g = 0$ (dirichlet) trovare un pezzettino che è un coseno

$f = 0$

$\vec{\beta}$ che ruota intorno al punto $(1,0) \rightsquigarrow \vec{\beta} = \begin{bmatrix} y-1 \\ -(x-0) \end{bmatrix}$